

[proposition]

Saliency as Contextual Budgeted Set Selection Under Uncertain Utility

Abstract

Many practical systems must choose a finite-cost subset of candidates when value depends on context, constraints, and interactions among the selected items. This paper formulates that problem as contextual budgeted set selection under uncertain utility. The true objective is latent expected downstream value, while the operational system optimizes a surrogate set utility that combines itemwise terms for novelty, goal-fit, long-run leverage, coherence, and temporal relevance with penalties for drag and redundancy and bonuses for complementarity. Hard safety, policy, validity, and compatibility constraints are represented as feasibility conditions rather than offsetting penalties. This yields a marginal-gain view of selection and clarifies the distinct roles of candidate generation, reranking, constrained optimization, control, and learning. Simpler saliency-style formulas appear as restricted special cases. Stronger optimality claims require additional structure, such as modularity or submodularity, and should not be stated without those assumptions.

1 Introduction

Finite-capacity selection recurs across retrieval, recommendation, memory management, routing, and attention support. A system faces more candidates than it can carry forward, each candidate consumes some resource, and the payoff of selecting a set depends both on context and on what else is selected. Many saliency-style formulations compress this into a single scalar item score. That compression is often useful as a mnemonic, but it is too coarse as a general theory.

The cleaner view is to treat saliency as a constrained set-selection problem. A chooser operates under finite capacity, hard feasibility constraints, uncertain downstream utility, and interaction effects among selected items. In that setting, the right formal object is not a universal item score, but a contextual set objective together with a feasible family and a selection procedure.

This reformulation makes four clarifications. First, it separates latent downstream value from the surrogate utility the system actually estimates and optimizes. Second, it places hard constraints in feasibility rather than in soft penalties. Third, it treats redundancy and complementarity as set properties rather than pretending they are intrinsic to isolated items. Fourth, it separates scoring from optimization, control, and learning. Those separations are structural, not cosmetic.

2 Formal Setup

2.1 Context, candidates, and feasibility

Let x denote the exogenous context for a single decision episode. Context may include user state, task, session history, device, time horizon, risk posture, and any other state that changes what is valuable now.

Unless stated otherwise, x is held fixed while the current selection problem is solved. If selection itself changes the relevant state, the model should be applied stagewise or lifted to a sequential decision framework.

Let $I(x)$ be the candidate universe available in context x .

Each candidate $i \in I(x)$ has:

- hard item-level eligibility $T_i(x) \in \{0, 1\}$,
- cost $c_i(x) > 0$ when cost-normalized scoring is used.

Some hard constraints are set-level rather than item-level. Let

$$C(A, x) \in \{0, 1\}$$

denote any additional hard feasibility condition on a set A , such as mutual exclusion, compatibility, quota, schema, or channel constraints.

Define the eligible universe

$$I^+(x) = \{i \in I(x) : T_i(x) = 1\}$$

and the feasible family

$$\mathcal{F}(x) = \left\{ A \subseteq I^+(x) : \sum_{i \in A} c_i(x) \leq B(x), C(A, x) = 1 \right\}.$$

This formulation keeps item-level gating and set-level gating distinct while allowing both to be enforced before or during selection.

2.2 Latent downstream value

Let

$$V^*(A, x) = \mathbb{E}[Y(A, x) \mid x]$$

denote the latent expected downstream value of selecting A in context x , where the expectation is over whatever uncertainty remains after conditioning on context.

The ideal chooser solves

$$A^*(x) \in \arg \max_{A \in \mathcal{F}(x)} V^*(A, x).$$

In practice V^* is not directly known. The system therefore estimates and optimizes a surrogate objective.

2.3 Remarks on zero-cost and sequential cases

If genuinely zero-cost items exist, ratio scoring by $1/c_i(x)$ is not automatically meaningful. Those items should either be handled by raw marginal gain, by a separate convention, or by a richer cost model.

If adding an item changes the context materially, write states $x_0, x_1, \dots$ and solve a sequence of stagewise selection problems or a fuller sequential decision problem. The present formulation addresses a single decision episode with context treated as fixed during selection.

3 Feature Vocabulary and Surrogate Utility

3.1 Stable feature vocabulary

For each candidate i , define:

- $N_i(x)$: novelty or information gain,
- $G_i(x)$: goal-fit or immediate relevance to the current task,
- $L_i(x)$: long-run leverage, such as reuse value, retention value, compounding value, or future usefulness,
- $H_i(x)$: coherence with the current trajectory, working set, or dialogue state,
- $\tau_i(x)$: age, staleness, or lag,
- $D_i(x)$: drag, meaning friction, interruption cost, cognitive burden, latency burden, or soft risk burden.

For pairs of candidates i, j , define:

- $K_{ij}(x)$: complementarity, synergy, or unlock value,
- $\text{Sim}_{ij}(x)$: redundancy, overlap, or substitutability.

Pairwise terms are taken to be symmetric unless otherwise stated. They are an approximation chosen for tractability, not a claim that all real interaction effects are pairwise.

3.2 Base item utility

Let θ collect the parameters of the surrogate model. Define the base item utility

$$u_i(x; \theta) = w_N(x) \phi_N(N_i(x); x) + w_G(x) G_i(x) + w_L(x) L_i(x) \\ + w_H(x) H_i(x) + w_F(x) \phi_F(\tau_i(x); x).$$

Here $w_*(x)$ are contextual weights, ϕ_N is a learned or chosen novelty response, and ϕ_F maps age or lag to temporal relevance.

This is the value of candidate i before interactions and soft burdens are applied. The weights should depend on context. Research mode, emergency mode, cold start, and long-session expert use should not all share the same tradeoff profile.

3.3 Set utility

Define the surrogate set utility

$$\hat{U}(A, x; \theta) = \sum_{i \in A} u_i(x; \theta) + \sum_{\substack{i < j \\ i, j \in A}} K_{ij}(x; \theta) - \lambda_{\text{div}}(x) \sum_{\substack{i < j \\ i, j \in A}} \text{Sim}_{ij}(x; \theta) - \sum_{i \in A} D_i(x; \theta).$$

This objective adds itemwise value, rewards complementarity, penalizes redundancy, and subtracts soft burdens.

Two distinctions matter. First, $c_i(x)$ is hard resource use that appears in feasibility. $D_i(x)$ is soft burden that appears in utility. They should not be conflated. Second, $\hat{U}$ is a set function for unordered selection. If order matters, the right object is a sequence function rather than a set function.

3.4 Operational problem

The deployed system solves

$$\hat{A}(x) \in \arg \max_{A \in \mathcal{F}(x)} \hat{U}(A, x; \theta),$$

or an approximation to that problem when exact optimization is too expensive.

This makes the separation explicit: $V^*(A, x)$ is the latent objective of interest, $\hat{U}(A, x; \theta)$ is the estimated operational surrogate, $\mathcal{F}(x)$ captures hard feasibility, and the selector is the algorithm used to approximate the argmax.

4 Marginal Gain and Operational Priority

The operational quantity for incremental selection is marginal gain relative to the already selected set A :

$$\Delta_i(A, x; \theta) = \hat{U}(A \cup \{i\}, x; \theta) - \hat{U}(A, x; \theta).$$

Under the pairwise surrogate above,

$$\Delta_i(A, x; \theta) = u_i(x; \theta) + \sum_{j \in A} K_{ij}(x; \theta) - \lambda_{\text{div}}(x) \sum_{j \in A} \text{Sim}_{ij}(x; \theta) - D_i(x; \theta).$$

Define the feasible marginal-density score

$$s_i(A, x; \theta) = \mathbf{1}[A \cup \{i\} \in \mathcal{F}(x)] \cdot \frac{\Delta_i(A, x; \theta)}{c_i(x)}.$$

This rule means: infeasible additions are excluded; candidates are scored by what they add now; and normalization by cost is used only when heterogeneous costs make such normalization meaningful.

If any feature depends on the evolving set rather than only on exogenous context, write that explicitly (for example $u_i(A, x; \theta)$), and compute Δ_i directly rather than relying on the pairwise expansion above.

5 Structural Propositions

The point of this section is to state exactly what follows from the formulation, and no more.

5.1 Basic set-function notions

A set function $f : 2^V \rightarrow \mathbb{R}$ is modular if $f(A) = \text{const} + \sum_{i \in A} a_i$. It is submodular if for all $A \subseteq B$ and all $i \notin B$,

$$f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B).$$

It is monotone if $f(A) \leq f(B)$ whenever $A \subseteq B$. Submodularity is the diminishing-returns property.

Proposition 1 (Itemwise ranking in the modular equal-cost regime). *Suppose that for fixed context x : (1) $\hat{U}(A, x; \theta) = \sum_{i \in A} v_i(x)$ is modular, (2) all eligible items have equal cost, and (3) feasibility is a cardinality constraint $|A| \leq k$. Then any set containing the k largest values $v_i(x)$ is optimal.*

Proof sketch. Under equal costs and a cardinality bound, feasibility depends only on the number of selected items. For any feasible set that omits a higher-valued item and includes a lower-valued one, exchanging the lower-valued item for the higher-valued one weakly increases the objective. Repeating the exchange yields a set of top- k values. $\square$

Proposition 2 (Density ranking fails under nonuniform costs). *Even when $\hat{U}$ is modular, ranking by value-per-cost is not generally optimal for 0–1 knapsack selection with nonuniform costs.*

Counterexample. Let $B = 50$ and three eligible items have (value, cost) pairs

$$a = (60, 10), \quad b = (100, 20), \quad c = (120, 30).$$

Ratios are 6, 5, 4, so density-first picks a then b , yielding value 160. But $\{b, c\}$ is feasible with value 220 > 160. $\square$

Proposition 3 (Redundancy-only penalties imply submodularity). *Suppose $K_{ij}(x; \theta) = 0$ for all pairs and $\text{Sim}_{ij}(x; \theta) \geq 0$. Define*

$$b_i(x; \theta) = u_i(x; \theta) - D_i(x; \theta)$$

and

$$f(A) = \sum_{i \in A} b_i(x; \theta) - \lambda_{\text{div}}(x) \sum_{\substack{i < j \\ i, j \in A}} \text{Sim}_{ij}(x; \theta).$$

Then f is submodular.

Proof sketch. For $A \subseteq B$ and $p \notin B$:

$$f(A \cup \{p\}) - f(A) = b_p - \lambda_{\text{div}} \sum_{j \in A} \text{Sim}_{pj},$$

$$f(B \cup \{p\}) - f(B) = b_p - \lambda_{\text{div}} \sum_{j \in B} \text{Sim}_{pj}.$$

Since $A \subseteq B$ and similarities are nonnegative, the second subtraction is at least as large, so diminishing returns holds. $\square$

Corollary 3.1 (Monotonicity requires an additional condition). *Under the assumptions of Proposition 3, the surrogate is monotone if and only if all feasible marginal gains are nonnegative.*

A convenient sufficient condition is

$$b_i(x; \theta) \geq \lambda_{\text{div}}(x) \sum_{j \in A} \text{Sim}_{ij}(x; \theta)$$

for every feasible addition of i to A .

Proposition 4 (Positive complementarity can break submodularity). *Suppose there exist distinct items a, b such that $K_{ab}(x; \theta) > 0$. Then the pairwise complementarity term is supermodular, and the full surrogate may fail to be submodular.*

Proof sketch. Consider

$$f(A) = \sum_{i \in A} b_i + K_{ab} \mathbf{1}[\{a, b\} \subseteq A].$$

With $A = \emptyset$, $B = \{b\}$, adding a gives marginal gains b_a versus $b_a + K_{ab}$. Since $K_{ab} > 0$, marginal gain increases after selecting b , violating diminishing returns. $\square$

Proposition 5 (Finite penalties cannot uniformly encode hard exclusion). *Suppose ineligibility is represented by subtracting a finite penalty $M > 0$ instead of using a hard gate:*

$$\tilde{U}(A, x) = U(A, x) - M \sum_{i \in A} (1 - T_i(x)).$$

Then for any fixed M , there exist instances in which an ineligible item is still selected.

Proof sketch. Pick an ineligible item q with $T_q(x) = 0$ that fits within budget. If $\Delta_q(\emptyset, x) > M$, then penalized marginal gain remains positive: $\Delta_q(\emptyset, x) - M > 0$. $\square$

6 Algorithmic Implications

The propositions above pin down what can and cannot be claimed.

In the modular, equal-cost, cardinality-constrained regime, top- k ranking is exact. That is the cleanest setting in which a single itemwise score really does solve the problem.

With modular utility and nonuniform costs, the problem becomes 0–1 knapsack. Even then, a density rule is only a heuristic, not a general optimum rule.

With redundancy-only pairwise penalties and nonnegative similarities, the surrogate becomes submodular. In that regime, diminishing returns appears naturally, and submodular-style algorithms become mathematically natural.

Once positive complementarity enters, the objective can become neither modular nor submodular. Then greedy procedures lose any blanket claim to optimality, and richer search or optimization procedures become appropriate. Depending on scale and latency budget, those may include beam search, local search, reranking loops, mixed-integer optimization, or learned approximate selectors.

The safe statement is therefore narrow:

Algorithmic guarantees depend on the structure of the surrogate objective and the feasible family. They do not follow from a salience score alone.

7 Architectural Decomposition

A practical system should separate six layers.

7.1 Eligibility layer

Apply hard feasibility checks. Item-level disqualifiers go into $T_i(x)$. Set-level disqualifiers go into $C(A, x)$.

Examples include safety rules, policy rules, schema validity, source validity thresholds, compatibility checks, mutual exclusion, and quota rules.

7.2 Candidate generation

Recall a broad, cheap pool from the candidate universe. This stage optimizes coverage and speed, not final precision.

7.3 Reranker

Estimate surrogate utility terms and interactions. This stage computes itemwise value signals, drag estimates, redundancy estimates, and complementarity estimates.

7.4 Selector

Choose the final feasible set under $\mathcal{F}(x)$. This stage is where greedy construction, beam search, local search, ILP, or other constrained optimization methods belong.

7.5 Controller

Adjust exploration, aggressiveness, budget tightening, temperature, channel multipliers, or other policy knobs that govern behavior across contexts. Exploration belongs here, not inside a single salience score.

7.6 Learner

Update θ , thresholds, transforms, interaction estimates, and controller policies from observed outcomes. Learning belongs here, not inside the same scalar equation used for online ranking.

This decomposition is useful because it prevents hard constraints, soft utility, exploration, optimization, and learning from being crammed into one overworked formula.

8 Learning and Evaluation

The separation between V^* and $\hat{U}$ has direct learning consequences.

Observed feedback typically arrives only for the selected set, not for all counterfactual feasible sets. That means learning the surrogate is partly a counterfactual estimation problem. A system may estimate component models for itemwise terms and pairwise terms, fit a direct predictor of set-level outcomes, or combine both approaches.

A generic learning target is to estimate parameters θ so that higher surrogate utility corresponds to better observed outcomes under deployment-relevant conditions. In symbolic form, one may fit θ by minimizing a loss of the form

$$\sum_t \ell(\psi(A_t, x_t; \theta), y_t) + \Omega(\theta).$$

where A_t is the selected set in episode t , y_t is the observed outcome, ψ is the model output used for training, and Ω is any regularizer or structural penalty.

Two evaluation cautions follow.

First, when interactions matter, itemwise ranking metrics can become a misleading proxy for decision quality. A system that ranks individual items well can still assemble poor sets.

Second, evaluation should track the level at which the system acts. If the selector chooses sets, then at least part of the evaluation should be set-level or policy-level rather than purely item-level.

This is one more reason the latent-versus-surrogate split is not just notation cleanup. It affects the learning target and the evaluation protocol.

9 Domain Instantiations

9.1 Retrieval and RAG

In retrieval settings:

- G captures query relevance,
- L captures downstream usefulness for reasoning or future reuse,
- H captures coherence with the current dialogue or working set,
- τ captures source age or lag,
- Sim captures chunk overlap,
- K captures multi-hop complementarity across evidence pieces,
- D captures token cost, latency, or distraction burden,
- T and C capture trust, source validity, safety, and policy constraints.

9.2 Training data selection

In training data selection:

- N captures undercoverage or surprise,
- G captures fit to the current training objective,
- L captures long-run capability gain,
- H captures curriculum coherence,
- τ captures distribution lag where relevant,
- Sim captures duplication or oversampling overlap,
- K captures complementary samples that unlock generalization,
- D captures labeling cost, noise burden, or instability risk,
- T and C capture license, validity, and policy constraints.

9.3 Product and engagement systems

In product flows:

- G captures immediate task completion value,
- L captures retention or future utility,
- H captures journey coherence,
- τ captures timeliness,
- Sim captures recommendation redundancy,
- K captures next-step unlocks or useful bundles,
- D captures friction and interruption burden,
- T and C capture trust, safety, and experience constraints.

9.4 Human operator mode

For human reasoning, a compact salience heuristic can still be useful. But it should be treated as an operator shorthand over this substrate, not as the substrate itself.

10 Scope of Claims

The model supports strong structural claims and weaker universal claims. Keeping those separate prevents puffed-up nonsense.

Safe claims:

- Many practical selection systems can be modeled as contextual feasible-set selection under uncertain utility.
- Itemwise salience formulas are restricted special cases of that broader problem.
- Hard exclusions are more robustly modeled as feasibility conditions than as finite penalties.
- Redundancy can produce diminishing returns, while complementarity can destroy them.

Unsafe claims unless extra assumptions are stated:

- One scalar item score explains all selection behavior.
- Greedy selection is optimal in general.
- Value-per-cost ranking solves the nonuniform-cost case.
- Hard constraints can always be replaced by sufficiently large soft penalties.

11 Conclusion

The useful general principle is not that one salience equation explains everything. The useful principle is that finite-capacity selection is a contextual feasible-set problem under uncertain utility. Candidates have costs. Some candidates are ineligible. Value depends on context. Redundancy and complementarity are properties of sets, not just of isolated items. The deployed chooser acts on an estimated surrogate through feasible marginal gains, often normalized by cost when that normalization is meaningful.

That framing preserves what was good in the original salience idea while removing notation drift, layer confusion, and fake universality.

A Compact Canonical Form

Given context x , define

$$\mathcal{F}(x) = \left\{ A \subseteq I^+(x) : \sum_{i \in A} c_i(x) \leq B(x), C(A, x) = 1 \right\}.$$

Latent downstream value:

$$V^*(A, x) = \mathbb{E}[Y(A, x) \mid x].$$

Surrogate utility:

$$\hat{U}(A, x; \theta) = \sum_{i \in A} u_i(x; \theta) + \sum_{\substack{i < j \\ i, j \in A}} K_{ij}(x; \theta) - \lambda_{\text{div}}(x) \sum_{\substack{i < j \\ i, j \in A}} \text{Sim}_{ij}(x; \theta) - \sum_{i \in A} D_i(x; \theta).$$

Ideal and operational problems:

$$A^*(x) \in \arg \max_{A \in \mathcal{F}(x)} V^*(A, x), \quad \hat{A}(x) \in \arg \max_{A \in \mathcal{F}(x)} \hat{U}(A, x; \theta).$$

Incremental priority:

$$\begin{aligned} \Delta_i(A, x; \theta) &= \hat{U}(A \cup \{i\}, x; \theta) - \hat{U}(A, x; \theta), \\ s_i(A, x; \theta) &= \mathbf{1}[A \cup \{i\} \in \mathcal{F}(x)] \cdot \frac{\Delta_i(A, x; \theta)}{c_i(x)}. \end{aligned}$$

B Legacy Mapping of S' , UEA, and SRC

Because exact legacy formulas are not reproduced in the visible thread, this mapping is structural rather than symbol-for-symbol.

B.1 S'

If S' was an itemwise scalar for ranking, it maps either to $u_i(x; \theta) - D_i(x; \theta)$ (raw item value) or to $s_i(A, x; \theta)$ (marginal-density priority). Under restrictive assumptions,

$$S'_i(x) \approx \frac{u_i(x; \theta) - D_i(x; \theta)}{c_i(x)}$$

when $A = \emptyset$, $K_{ij} = 0$, $\text{Sim}_{ij} = 0$, $T_i(x) = 1$, and set-level feasibility is trivial.

B.2 UEA

If UEA denoted true downstream performance, it maps to $V^*(A, x)$. If it denoted operational scoring, it maps to $\hat{U}(A, x; \theta)$ or one of its components. Those two roles should not share one symbol.

B.3 SRC

If SRC governed exploration, aggressiveness, routing, or channel choice, it belongs to the controller layer. If it performed constrained feasible-set choice, it belongs to the selector layer. It should not be folded as another additive utility term.

B.4 Trust/safety modifiers

Binary disqualifiers should move into $T_i(x)$ or $C(A, x)$. Graded source-quality effects can remain soft terms but should be represented explicitly as such.

B.5 Freshness/novelty multipliers

Fixed multipliers map to contextual weights $w_N(x)$, $w_F(x)$ or response functions $\phi_N(\cdot; x)$, $\phi_F(\cdot; x)$, separating importance weighting from response shape.

C Drop-in Claim Language

1. **Structural framing.** Many retrieval, recommendation, routing, and memory systems can be modeled as contextual feasible-set selection under uncertain utility.
2. **Operational framing.** A useful strategy is to estimate feasible marginal utility, often normalized by cost, using a surrogate with drag, redundancy, and complementarity.
3. **Special-case framing.** Itemwise salience scores are exact only in restricted regimes where the objective is effectively modular and the feasible family is simple enough to collapse set selection into ranking.

4. **Constraint framing.** Hard policy, safety, and validity constraints are better represented as feasibility conditions than as tradeable penalties.
5. **Guarantee framing.** Approximation or optimality claims require explicit assumptions such as modularity, monotonicity, or submodularity.